SUMMARY OF THE INVENTION

The present invention may be implemented into a system and method for communicating, or storing and retrieving, a compressed digital representation of a document. The input document is first digitized, and the system will perform a graphics format packing procedure according to either an automatic or manual mode of operation. For the automatic mode, the system will determine the maximum bit representation of the document and will send the information to a byte packing operation; for the manual mode, a dithering technique will be used to dither the document to a bit map with the exact bit representation required by the user. Byte packing compression is then performed on a row-by-row basis, in effect converting the digital representation into a Portable Grey Level (PGM) format. The PGM format document is then decomposed according to a wavelet analysis technique, followed by lossless compression if desired. The compressed decomposed document may then be stored or communicated over conventional communication networks. Decompression and reconstruction of the document is performed by reversal of the compression and decompression processes, at the receiving end, for display of the document.

The present invention is based on finite elements, or splines. Hence, although the operations are in digital formats, the results are in an analog format, such that information can be extracted from anywhere in the result to improve the display resolution with real-time efficiency. In addition, the use of boundary wavelets according to the present invention eliminate boundary effects. Each of these two attributes separate the present invention from conventional compression technology.

Furthermore, according to the present invention, binary operations may be used to advantage at any stage of the process. This is made possible from the use of integer coefficients in the decomposition and reconstruction filters. According to one of the alternative embodiments, the use of dual bases is also accomplished by integer sequences filters, and allows the user to choose a shorter, and computationally more efficient, filter at either or both of the compression or decompression ends. A second alternative embodiment, namely interpolatory wavelets, provides highest efficiency compression and decompression, while a third alternative embodiment, namely wavelet packets, provides the highest quality compression and decompression by way of full-tree decomposition and reconstruction.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an electrical diagram, in block form, of a communications and storage system according to the preferred embodiment of the invention.

Figure 2 is a flow diagram illustrating the operation of the system of Figure 1 in compressing document data according to the preferred embodiment of the invention.

Figure 3a is an electrical diagram, in block form, of the format converter of the system of Figure 1, and Figures 3b and 3c are flow diagrams illustrating the automatic and manual operation modes, respectively, of the format converter of Figure 3a, according to alternative embodiments thereof.

Figure 4 is an electrical diagram, in block form, of the spline-wavelet compressor of the system of Figure 1, according to the preferred embodiment of the invention.

Figures 5a and 5b are tree diagrams illustrating the decomposition and reconstruction of a signal function according to spline-wavelet techniques.

Figure 6 is a graphical illustrations of the cardinal B-spline scaling function utilized according to the preferred embodiment of the invention.

Figure 7 is a flow diagram illustrating the general procedure for spline-wavelet decomposition of a document according to the preferred embodiment of the invention.

Figure 8 is a flow diagram illustrating the procedure for spline-wavelet decomposition of a document according to a first alternative implementation of the invention, specifically the dual-base technique.

Figure 9 is an illustration of the operation of the process of Figure 8.

Figures 10a and 10b are plots of the scaling function and wavelet function utilized according to a second alternative implementation of the invention, specifically the interpolatory wavelet technique.

Figures 11a and 11b are full-tree representations of the decomposition and reconstruction, respectively, of a signal function according to spline-wavelet techniques, including a third alternative implementation of the invention utilizing wavelet packets.

Figures 12a and 12b are flow diagrams illustrating the procedure for spline-wavelet decomposition of a document according to the third alternative implementation of the invention, specifically the wavelet packets technique.

Figures 13a through 13e illustrate the operation of the process of Figures 12a and 12b on an array of coefficients.

Figure 14 illustrates the operation of the process of Figures 12a and 12b in a tree diagram form.

Figure 15 illustrates the organization of a data frame containing wavelet packet decomposition results and an associated header containing a decomposition tree map.

Figure 16 illustrates a decomposition tree map according to the wavelet packet technique, illustrating the effects of pruning at certain component sequences.

Figure 17 is a flow diagram illustrating the general procedure for spline-wavelet decomposition of a document according to the preferred embodiment of the invention.

Figure 18 is an electrical diagram, in block form, of the spline-wavelet decompressor of the system of Figure 1, according to the preferred embodiment of the invention.

Figure 19 is a flow diagram illustrating the procedure for spline-wavelet reconstruction of a document according to a first alternative implementation of the invention, specifically the dual-base technique.

Figure 20 is a flow diagram illustrating the procedure for spline-wavelet decomposition of a document according to a second alternative implementation of the invention, specifically the interpolatory wavelet technique.

Figure 21 illustrates the reconstruction tree for spline-wavelet reconstruction of a generalized document according to the wavelet packets process.

Figure 22 is a flow diagram illustrating the procedure for spline-wavelet reconstruction of a document according to the wavelet packets technique.

Figure 23 illustrates the operation of the process of Figure 22 in a tree diagram form, for an exemplary document.

DETAILED DESCRIPTION OF THE INVENTION

1. Document communication and storage system with compression and decompression

Referring now to Figure 1, a system for communicating or storing documents according to the preferred embodiment of the invention will now be generally described. The exemplary system shown in Figure 1 includes compression system 10 and decompression system 30 in communication with one another, such as via digital communication network 26. As such, this embodiment of the invention will be described with respect to these discrete compression and decompression locations; it is to be understood, of course, that the same physical computer or location may contain, or serve as, both compression system 10 and decompression system

30, particularly in the application of an archival system using the compression techniques to be described herein.

In the embodiment of the invention illustrated in Figure 1, compression system 10 receives digital information representative of a document from image source 8. As noted hereinabove, the term "document" as used in this application refers to a two-dimensional representation or rendering of a nature that is conventionally made on a paper medium or on a graphics display by writing, printing, typing, drawing, operation of a computer-aided draw, paint or design program, or other similar conventional techniques, and those renderings that are commonly stored on paper, microfilm, microfiche, or electronically. As noted above, examples of such documents include text documents, bank checks and other banking transaction records, vital and legal records, maps, charts, printed works (including combinations of text and graphics), seismic plots, medical records, bank and insurance records, directories, identification images (pictures, signatures, etc.) and the like. As is well known to the public, examples of conventional electronic communication and storage of documents includes modern digital facsimile ("fax") equipment, CD-ROM storage and distribution of encyclopedias and other series of books, electronically or magnetically stored representations of transactions such as checks and other banking transactions and statements, and the like. As such, image source 8 may be implemented as a document scanner or a facsimile machine, in the case where the document is a paper or other hard copy representation. Alternatively, if the document is already in electronic form, image source 8 may be implemented as a computer system that receives the document from another electronic (or graphics) format or from another computer system over a communications

network, that retrieves the document from its own disk or tape storage, that receives the document directly via a fax modem, or that is used to generate the document itself. Of course, it should be apparent to those of ordinary skill in the art that other conventional image sources may alternatively serve as image source 8 in the system of Figure 1.

The output of image source 8 is preferably a digital data stream representative of the appearance of the document, such as a bit map representation. Compression system 10 includes format converter 12 and spline-wavelet compressor 20. Format converter 12 receives the output of image source 8 (in the general case) and converts the digital representation received from image source 8 into a full ASCII binary bitmap representation suitable for compression by spline-wavelet compressor 20. According to the preferred embodiment of the invention, as will be described in detail hereinbelow, format converter 12 converts the input document into a binary ASCII form expressed in Portable Grey Level (PGM) format. Spline-wavelet compressor 20 performs a lossy compression of the input document, and formats the compressed document into the appropriate output format, in the manner described hereinbelow.

The compressed representation of the input document may be communicated by spline-wavelet compressor 20 in compression system 10 over communications network 26 to a receiving computer or similar hardware receiving device including decompression system 30. Communication network 26 may be a conventional analog transmission or electronic digital communications network, or both an analog and digital network when including the appropriate clusters, analog-to-digital and digital-to-analog converters, and other necessary apparatus. Network 26

may be realized according to any conventional technology, including telephone line (including T1, T2, and T3 communication), hard-wired cable, fiber optic cable, broadcast or satellite transmission, and the like. Alternatively or in addition, the output of compressor system 10 may alternatively be presented to bus 25 for storage at main disk 22 (e.g., for archival storage), or to floppy disk 24 for off-line storage or for transmittal to another computer; of course, storage on other media such as tape and the like is also contemplated.

Decompression system 30 according to the system of Figure 1 is operable to receive the compressed document data directly from compression system 10 over communications network 26, or via computer bus 27 from disk storage 22 or floppy disk 24. It will be apparent to those of ordinary skill in the art that the actual communication of the compressed representation of the document may, of course, be performed by way of a single transmission between compression system 10 and decompression system 30, or alternatively by way of a broadcast transmission by compression system 10 to multiple decompression systems 30.

Many other types of transmissions are, of course, also contemplated. For example, compression system 10 may store the compressed document onto archival disk storage 22, for example at a central server unit or location, such as a bank, library, central company office, or document service. In this case, many documents will be compressed and stored for later communication. Decompression system 30 may then call this central location to access one or more documents, in which case the central location will retrieve the appropriate compressed documents from its main disk 22 and communicate the same to decompression system 30 over

communications network 26. Further in the alternative, a remote computer may request certain documents, in compressed form, from a central storage location; after receipt of the compressed documents via communications network 26 or by transport of the compressed documents on portable floppy disk 24, the remote requesting computer may not immediately forward the compressed documents to decompression system 30, but instead store the documents, in compressed form, in its main disk 22. Decompression system 30 will then receive the compressed documents via its computer bus 27 for decompression as desired.

It is therefore contemplated that the system of Figure 1 may be implemented or operated in these and other similar ways within the scope of the present invention, in the fields of data and record communication, record archival and retrieval, catalog services, and the like. In addition, it is contemplated that the system of Figure 1 will also be useful in providing services such as on-demand video images in the nature of "picture book" services, the transmission of still images such as useful with video telephonic equipment, identification systems in which pictures and signature images are communicated, and the like.

In any case, decompression system 30 includes spline-wavelet decompressor 40 and format converter 32. Spline-wavelet decompressor 40 receives the compressed document data, and performs substantially the reverse of the process performed by spline-wavelet compressor 20 in compressor 10. The output of spline-wavelet decompressor 40, according to this preferred embodiment of the invention, will be a PGM format representation of the compressed document. As will be described hereinbelow, since the compression technique according to the present invention is of the lossy type, the output of

decompressor 40 in decompression system 30 may not be identical to the input to compressor 20 in compression system 10. The output of spline-wavelet decompressor 40 is presented to format converter 32, which reformats the decompressed document into the suitable form for display on video display 34d, for printing by printer 34p, or for output by way of another conventional output device.

2. The compression system

The construction and operation of compression system 10 according to the preferred embodiment of the invention will now be generally described. Compression system 10 may be a stand-alone system, for example as a special archiving machine dedicated for document scanning and storing, or can be arranged so as to be an add-on card for a conventional high performance personal computer or workstation; of course, compression system 10 may also be implemented as a function of a larger computer, such as a mainframe computer or supercomputer. As illustrated in Figure 2, the compression process begins with process 42, in which format converter 12 receives a digital bit stream representative of the input document from image source 8. The format of the digital bit stream representative of the document, as produced by image source 8 and presented to format converter 12, may be any one of the known formats such as PCX, IMG, GIF, TIF, RLE, NTSC, PAL, and the like. Compression system 10 according to this embodiment of the invention is able to compress input documents of various types, not only monochromatic (two-color) documents, but also documents that are in full color or in a grey-scale form. Format converter 12, as noted above, receives this bit stream from image source 8 and converts it to a binary PGM format, as will now be described in detail.

a. Format conversion

Upon receipt of the digital document in process 42, format converter 12 performs process 44 of Figure 2, in which the digitally-represented received document is transformed into a byte-packed PGM format for compression by spline-wavelet compressor 20.

Referring now to Figure 3a, the construction of format converter 12 will now be described in detail. As shown in Figure 3a, format converter 12 according to this embodiment of the invention includes format decoder and color expansion device 14, which has an input for receiving the input data stream from image source 8. Format converter and color expansion device 14 is preferably implemented in the conventional manner for performing the decoding and expansion operations described hereinbelow. For example, circuitry for performing such format decoding is well known in the art, such as described in Rimmer, Supercharged Bitmapped Graphics (Windcrest/McGraw Hill 1992); conventional circuitry for performing such color expansion is also readily available, and includes such devices as color palette tables in RAM, ROM or in other electronic storage devices.

The output of format converter and color expansion device 14 is forwarded to bit determination device 15, and also to symbolic substitution circuitry 16, to which the output of bit determination device 15 is also presented. Symbolic substitution circuitry 16 presents its output to byte packer 18, which completes the format conversion as necessary prior to the compression operation according to this embodiment of the invention, for presentation to spline-wavelet compressor 20. Each

of bit determination device 15, symbolic substitution circuitry 16, and byte packer 18 may be constructed according to conventional circuitry for performing these operations, or alternatively may be performed by a single high performance graphics or general purpose processor, as desired.

Referring now to Figures 3b and 3c, the operation of format converter 12 in performing transformation process 44 will now be described in detail. Figures 3b and 3c represent manual and automatic modes of transformation process 44, respectively.

Referring first to Figure 3b, relative to the manual mode of transformation process 44, transformation process 44 begins with process 39, in which format converter 12 receives a user input regarding the number of bits by which each pixel of the document or image is to be represented. In process 41, format decoder and color expansion device 14 decodes the data from the format presented by image source 8, such format being PCX, IMG, TIF, GIF, RLE, YUV, etc., into a color matrix, or color palette table, representation of the image signal. It is preferred that format decoder and color expansion device 14 also be able to detect the type of document to which the input bit stream corresponds, to provide flexibility in the types of documents presented to compression system 10. In the case of color input documents, format decoder and color expansion device 14 may thus also, in process 41, perform a color expansion operation so that the output from device 14 is in an expanded RGB format, with each pixel represented by red, green and blue color component bytes.

Format decoder and color expansion device 14 preferably also includes sufficient logic to perform

decision 43 (of Figure 3b), in which the determination is made as to whether the input document is a "satisfied" document, that is whether the number of bits representing each pixel in the input document is less than or equal to the number of bits specified in process 39. For example, in the case of color documents, the output of format decoder and color expansion device 21, after the color expansion operation, may correspond to digital values in an RGB format, where each color component is represented by a multiple-bit value indicative of the intensity of that color. Similarly, each pixel in a representation of a grey-scale document will have a single color multiple-bit component corresponding to the intensity of the brightness (in a black and white system) for that pixel, while a representation of a two-color (e.g., black and white), or binary, document requires only a single bit per pixel. If decision 43 determines that the output of format decoder and color expansion device 14 satisfied the user specification, control passes to process 47, which is described hereinbelow.

However, if decision 43 determines that the input document does not satisfy the user specification (i.e., the number of bits per pixel in the input document exceeds the number of bits specified by the user in process 39), the output of format decoder and color expansion device 14 is forwarded to bit determination device 15, to perform the desired dithering process necessary to convert the digital representation of the document to the exact bit representation. As is well-known in the art, for example, the process of dithering corresponds to displaying a grey color in a monochrome representation where the shade of grey depends upon the relative density of black and white pixels for the corresponding portion of the image. The Rimmer reference cited hereinabove also provides a description of

conventional dithering techniques. It is contemplated that conventional circuitry may be used for bit determination device 15, including conventional graphics processing circuitry.

According to the preferred embodiment of the invention, bit determination device 15 is controllable so as to operate according to one of several dithering modes. For example, in the case where bit determination device 15 is implemented as a programmable graphics processor, it is contemplated that bit determination device 15 may include program code corresponding to each of several dithering routines, selectable by way of a pointer or the like. For example, Table 1 hereinbelow illustrates the correspondence between a dithering mode code and the selected dithering method, according to this embodiment of the invention:

Table 1

Dithering Mode Code	Dithering Method
1	Bayer-Dickering Method
2	Floyd-Steinberg Method
3	Burke's Method
4	Stucki Method
0	Dithering disabled

As a result of dithering process 45, or alternatively if the output of format decoder and color expansion device 14 already corresponds to a proper color representation (as determined by process 43), the symbolic substitution process 46 and byte packing process 47 described hereinbelow for the binary and non-binary documents, as the case may be, is then performed.

Referring now to Figure 3c, the automatic mode of format conversion process 44' will now be described in detail. Process 44' begins with color expansion of the document in process 41, as before. The hardware construction of format converter 12 for this embodiment of process 44' will be similar as that shown in Figure 3a, and as such no additional description of the circuitry will be provided herein. For non-two-color (i.e., color or gray-scale) documents, the output of color expansion process 41 will be an image table with at least one byte of data per pixel. Following color expansion process 41, process 49 is performed in this automatic mode to determine the number of colors necessary to represent the document is selected. Process 49 may be done by way of a color histogram statistic method, in which the number of colors necessary to represent the document is selected by determining the frequency of true colors in the document and by determining how many are reasonably necessary to capture the bulk of the information. Process 49 thus results in each byte (or larger word) for each pixel containing one of a fixed set of values, representative of a reduced number of colors, associated with that pixel. Process 49 may alternatively be performed by way of a selected dithering technique, to represent the image by a reduced number of colors. In either event, the image table after process 49 will be a representation of the image by a reduced number of colors, represented by a reduced set of values (a, b, c, . . .).

Whether produced by the manual or automatic format conversion processes 44, 44' described hereinabove, the result will be an image table having a digital representation of intensity or color for each pixel. Symbolic substitution process 46 and byte-packing process 47 are then performed on this image table, as shown both

in Figures 3b and 3c. However, symbolic substitution process 46 and byte-packing process 47 will differ for binary document than for multi-color documents, as will be described hereinbelow.

i. Symbolic substitution and byte-packing for binary documents

In the case of a binary document (i.e., a two-color, black and white, document), the document is represented as an image matrix or table containing two values a, b, corresponding to black and white, respectively. For example, in the case where each pixel in the document is represented by an eight-bit value, the value "0" may represent black and the value "255_{10}" may represent white; of course, the dithering process may assign different numerical values to the two colors represented. Symbolic substitution process 46, performed by symbolic substitution circuitry 16 of format converter 12, is operable to detect the two values a,b in the image table, and to substitute the binary values 0,1 therefor, respectively.

For example, in the case of an eight-by-eight image table, the output from bit determination device 15 (or, if appropriate, directly from format decode and color expansion device 14), may appear as follows:

a	b	a	a	a	b	b	b
b	a	a	b	b	a	a	b
a	a	a	a	b	a	b	a
b	b	a	b	a	b	b	a
a	a	a	b	b	a	a	b
b	a	b	a	a	a	b	a
a	a	b	b	a	b	a	a
b	b	a	a	b	a	b	b

In the case where each pixel is represented by an byte, and where the two color codes a, b correspond to 0, 255, respectively, this image table would appear as follows:

0	255	0	0	0	255	255	255
255	0	0	255	255	0	0	255
0	0	0	0	255	0	255	0
255	255	0	255	0	255	255	0
0	0	0	255	255	0	0	255
255	0	255	0	0	0	255	0
0	0	255	255	0	255	0	0
255	255	0	0	255	0	255	255

After symbolic substitution process 27, this image table would appear as follows:

0	1	0	0	0	1	1	1
1	0	0	1	1	0	0	1
0	0	0	0	1	0	1	0
1	1	0	1	0	1	1	0
0	0	0	1	1	0	0	1
1	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
1	1	0	0	1	0	1	1

Of course, if the two color codes a, b from bit determination device 15, or from format converter and

color expansion device 14, were already 0, 1, respectively, symbolic substitution process 46 would be unnecessary.

For this example of a binary document, once the image table for the document is converted to binary values, byte packing circuitry 18 performs byte packing process 47 to place the document into a PGM format. Byte packing process 47 converts each eight binary bits in the image table, representative of eight adjacent pixels in the document, into an eight-bit word (byte) for subsequent numerical processing. Process 47 thus assigns binary powers to each of the pixel locations in an eight-pixel group; for example, the left-most pixel may be assigned the power 2^7 and the right-most pixel in the group may be assigned the power 2^0, with the pixels in between corresponding to the intermediate powers in order. The resultant byte-packed representation of the eight-bit pixel groups then corresponds to the sum of the values in each pixel location (1 or 0) times the assigned power of two for that location. The exemplary eight-by-eight image table noted above would appear, after byte-packing in the horizontal direction, as follows (all numbers decimal):

71
153
10
214
25
162
52
203

Of course, horizontally adjacent eight-pixel groups in a larger document would have their own byte-packed values.

If the horizontal size of the document is not a multiple of eight, the remainder bits are "zero-packed". For example, if the horizontal size of the input document is 70 pixels, two zeroes may be added at the end of the pixel string prior to the byte-packing process; since the actual size of the document is known, such zero-packing will not result in any error or confusion. Byte-packing process 47 thus reduces the storage capacity necessary for representation of a two-color document significantly; for example, where the two-color image table represents each pixel with a byte, byte packing process compresses the image table by a factor of eight. The result of byte packing process 47 thus resembles an eight-bit PGM format, which is particularly convenient for processing by microprocessors or computers using ASCII coding and programming environments, as the fundamental storage unit is the byte in these environments.

<u>ii. Symbolic Substitution and byte-packing for non-binary documents</u>

While the foregoing description of byte-packing according to the present invention was directed to binary (i.e., two-color) documents, it is of course well known that a wide range of documents are not of two-color type, and thus cannot be represented in binary form without the loss of information. For example, conversion of color and gray-scale documents into binary form will necessarily cause the loss of the color and gray-scale information, respectively. In addition, even if the original input document is a two-color document, practical limitations in the scanning or digitization hardware used to convert the document into machine-readable form may result in a digital representation that may not be two-color in the strictest sense, in which

case restriction of the output into binary form will also cause the loss of information. These effect limit not only documents that are fully in graphics form, but also the graphic portion of documents containing both text and graphics.

It is therefore useful to digitally represent these documents with more than a single bit, for example with two, three, four or more bits, depending upon the range of "color" information to be considered in the process. As is fundamental in the art, to digitally represent n colors in a document, 2^n bits of information are required for each pixel; for example, an image having up to sixteen colors requires four bits per pixel in its digital representation, and an image allowing up to eight colors requires three bits per pixel. However, as in the case hereinabove for the two-color document, an entire byte is generally used for the representation of each pixel, even though only eight colors are necessary. Symbolic substitution and byte packing according to this embodiment of the invention will be able to save significant memory space, thus providing a significant degree of lossless compression.

For example, a document may be represented by eight colors per pixel as a result of decision 43 or process 45 in the manual format conversion mode, or of process 49 in the automatic format conversion mode, whether corresponding exactly to the input document, generated by color histogram statistics or generated by dithering. In this example, the eight colors are represented by an eight bit value (i.e., ranging from 0 to 255_{10}), and consist of the set (0, 24, 53, 75, 103, 148, 196, 255); in hexadecimal notation, this set would be (00, 18, 35, 4B, 67, 94, C4, FF). An example of an image table for an

eight color document of a size eight pixels by eight pixels is as follows:

53	196	148	103	24	24	75	53
255	196	196	103	53	24	255	103
148	196	75	53	24	24	0	0
75	53	53	255	103	148	0	24
75	53	255	24	0	148	148	103
196	148	24	24	0	196	53	53
24	255	53	196	75	148	103	0
24	255	196	196	255	75	53	75

Process 46 is then performed in which a symbolic substitution is performed such that each of the reduced number of colors is represented by the smallest (in number of bits) word size. For example, if the result of process 49 is a set of eight color values over the range from 0 to 255_{10}, the symbolic substitution of process 46 represents these eight values by the eight binary values 0 through 111_2, stored in each byte (or larger word) associated with each pixel. For the exemplary image table shown above, the symbolic substitution of process 46 would perform the following substitution:

8-bit Value (hexadecimal)	Substituted Value
00	00
18	01
35	02
4B	03
67	04
94	05
C4	06
FF	07

The resultant exemplary image table, after the symbolic substitution of process 46, would appear, in hexadecimal form, as follows:

02	06	05	04	01	01	03	02
07	06	06	04	02	01	07	04
05	06	03	02	01	01	00	00
03	02	02	07	04	05	00	01
03	02	07	01	00	05	05	04
06	05	01	01	00	06	02	02
01	07	02	06	03	05	04	00
01	07	06	06	07	03	02	03

As indicated by the above table, after the symbolic substitution of process 46, each pixel is still represented as a full byte.

It has been found, according to this embodiment of the invention, that byte packing can be done in such a manner as to ignore pixel boundaries, such that the color information for a pixel may be partially contained in one of the packed bytes and partially contained in the next packed byte in sequence. After the symbolic substitution of process 46, byte packing process 47 according to this embodiment of the invention is then performed to reduce the number of bits required to store the image table to the minimum required.

For the example described hereinabove, it is seen that a full byte is used to represent eight possible values, which at most require three bits; the preceding table thus includes five unnecessary bits for each pixel. Process 47 thus may be considered to include the operation of discarding the excess bits, such that only three bits per pixel (in this example) are used. The

exemplary image table may thus be expressed, in binary form, as follows:

010	110	101	100	001	001	011	010
111	110	110	100	010	001	111	100
101	110	011	010	001	001	000	000
011	010	010	111	100	101	000	001
011	010	111	001	000	101	101	100
110	101	001	001	000	110	010	010
001	111	010	110	011	101	100	000
001	111	110	110	111	011	010	011

According to this embodiment of the invention, this image table may be packed into eight-bit bytes by regrouping the bits in this table into groups of eight, rather than into groups of three as used to represent the eight colors of the image. This regrouping of the digital information without regard to pixel boundaries is made possible by the subsequent spline-wavelet operations, in which the image data is transformed; as such, it has been discovered that pixel boundaries need not be maintained in the byte-packing process. Of course, if the number of colors of the document is such that the number of bits per pixel evenly divides into eight (i.e., one, two or four), the pixel boundaries will match up with the byte boundaries.

With reference to the eight-color example above, byte packing by process 47 will produce an image table in which the first byte in each row will contain the first two pixels plus the two most significant bits of the third pixel; the second byte will contain the least significant bit of the third pixel, the fourth and fifth pixels, and the most significant bit of the sixth pixel; and the third byte will contain the two least significant bits of the sixth pixel, and the seventh and eighth

pixels. In general, additional zero bits may be added in order to arrive at full bytes, depending upon the number of pixels in the image. The exemplary image table noted above will appear as follows upon completion of byte packing operation 46':

01011010	11000010	01011010
11111011	01000100	01111100
10111001	10100010	01000000
01101001	01111001	01000001
01101011	10010001	01101100
11010100	10010001	10010010
00111101	01100111	01100000
00111111	01101110	11010011

Represented in hexadecimal notation, the image table would be represented as:

5A	C2	5A
FB	44	7C
B9	A2	40
69	79	41
6B	91	6C
D4	91	92
3D	67	60
3F	6E	D3

In this example, therefore, process 44' has compressed the sixty-four original bytes into twenty-four bytes, providing a compression ratio of 2.667 prior to decomposition. The compression ratio obtained by process 44' according to this embodiment of the invention will, of course, increase to the extent that fewer colors are necessary for the representation of the image; of course, the resultant image will be nearer the input document as more colors are allowed.

Upon completion of byte-packing process 47 (or 47', as the case may be), transformation process 44 (or 44') of Figure 2 is thus completed, with the results forwarded by format converter 12 to spline-wavelet compressor 20 in PGM format which, as will be seen, is quite convenient for the compression operations to be described below. In addition, as is evident from the foregoing description, format converter 12 according to this embodiment of the invention is operable to convert a wide range of input document formats, including full color or gray scale representations, into a PGM format, with a significant amount of compression already achieved. For example, a 1024-by-1024 (pixels) image table reduces to a 128-by-1024 table for compression by spline-wavelet compressor 20.

b. Spline-wavelet compression

Referring back to Figure 2, following completion of transformation process 44, spline-wavelet compressor 20 performs process 48, in which the byte-packed PGM format representation of the input document is decomposed; decomposition of the image information is an integral part of the compression of the document for transmission or storage. According to the present invention, the decomposition of process 48 may be performed according to one of three processes 48a, 48b, 48c (collectively referred to as process 48) utilizing spline-wavelet decomposition, each of which will be described in detail hereinbelow. As will become apparent, these alternative spline-wavelet decomposition processes 48 allow selection of the appropriate compression technique depending upon the performance of the compression and decompression systems, the desired compression ratio, the desired compression or decompression speed, and other parameters.

According to the preferred embodiment of the invention, compression system 20 is multiply programmed so that the user may select the specific decomposition one of processes 48a, 48b, 48c.

In the general sense, decomposition process 48 is followed by quantization of the wavelet components of the decomposed image signal, in process 50; this quantization enables compression of the storage requirements for the input document, by eliminating from the information those components of the decomposed image that are unimportant to the image (as determined by the decomposition process). Decision 51 is then performed by compression system 20 to determine if the compression ratio after quantization process 50 is acceptable; if not, decomposition process 48, for additional levels, and quantization process 50 are again performed to further compress the image information. Upon the desired compression ratio being reached (as determined by decision 53), lossless compression may be performed by compression system 20 in process 52 according to conventional techniques, following which compression system 20, in process 54, formats the compressed image information into the format appropriate for communication over digital network 26 or for archival storage by way of disk storage 22, 24.

<u>i. Construction of spline-wavelet compressor</u>

Referring now to Figure 4, the construction of spline-wavelet compressor 20 according to the preferred embodiment of the invention will now be described in detail. Spline-wavelet compressor 20 includes data controller 56, which receives the PGM formatted image information from format converter 12 and which also,

according to this example, controls the operation of the other components within spline-wavelet compressor 20. In this example, data controller 56 presents and receives signals on control bus 63 to control the timing, feedback and transmission of information through and from spline-wavelet compressor 20, including the case where data received from format converter 12 is transmitted directly through spline-wavelet compressor 20 without compression. Data controller 56 is also preferably able to detect the presence of input data presented by format converter 12, and to detect the format or type of data so presented. In addition, data controller 56 also preferably receives control signals, for example from the computer into which compression system 10 is installed as an add-on function; these control signals may indicate the type of compression desired, the type of quantization to be performed, and the selection of other options available in the compression process, as will be described in detail hereinbelow. It is therefore contemplated that data controller 56 may be implemented as a relatively simple logic circuit, for example as implemented into a gate-array or other semi-custom logic circuit, for performing these functions.

In this embodiment of the invention, data controller 56 forwards the input data that it receives from format converter 12 to digital signal processor (DSP) 60 by way of data bus 58. DSP 60, as will be described in further detail hereinbelow, is the main processing unit for performing spline-wavelet decomposition process 48 (Figure 2). Examples of a readily available digital signal processor suitable for use as DSP 60 according to this embodiment of the invention are the TMS320C25 and TMS320C30 digital signal processors manufactured and sold by Texas Instruments Incorporated; of course, other digital signal processors and microprocessors may

alternatively be used to perform the spline-wavelet decomposition operations described hereinbelow. DSP 60 is coupled to data bus 61d and address bus 61a (collectively referred to as memory bus 61) for communication with program memory 65 and image memory 67. According to this embodiment of the invention, DSP 60 is programmed (by way of code stored within DSP 60 and also in program memory 65) to perform decomposition of the data in PGM format presented to it by data controller 56 on bus 58 according to pre-calculated decomposition coefficients stored in program memory 65. The image data is stored in image memory 67 before, during and after the decomposition; in this example, image memory 67 is arranged in four banks, totaling two megabytes in capacity. DSP 60 is also connected to control bus 63, so that it may be controlled by data controller 56 and so that it can effect the necessary control of other components in spline-wavelet compressor 20, including memory 65, 67, during the performance of decomposition process 48, described in further detail hereinbelow.

As noted above, according to this embodiment of the invention, spline-wavelet decomposition process 48 may be performed by DSP 60 according to one of several optional decomposition or compression modes, controlled by data controller 56. These modes may be selected by a code communicated by data controller 56 to DSP 60 over control bus 63, an example of which is as follows:

0	:	No compression
1	:	Dual base wavelet decomposition
2	:	Interpolatory wavelet decomposition
3	:	Wavelet packet decomposition
4-7	:	Reserved

Each of the three decomposition techniques, and the resulting compression of the input document, will be described in detail hereinbelow.

DSP 60 is also bidirectionally coupled to quantization processor 62, to which DSP 60 forwards the results of the decomposition process. Quantization processor 62 quantizes the decomposed information (process 50 of Figure 2) and forwards the same to lossless compressor 64 or back to DSP 60, depending upon the results of decision 51 in determining whether the desired compression ratio has been reached. The details of quantization process 50 will be described in further detail hereinbelow. According to this embodiment of the invention, quantization processor 62 is preferably implemented as a programmable microprocessor or custom logic circuit for performing the functions described hereinbelow; such implementation is believed to be readily apparent to one of ordinary skill in the art having reference to this description.

Quantization process 50 may be performed by quantization processor 62 according to one of various selectable conventional quantization modes, under the control of data controller 56 via control bus 63. These modes may be communicated to a three-bit quantization mode register within quantization processor 62, which controls the type of quantization to be performed. An example of the quantization codes and their corresponding quantization modes are as follows:

0	:	No quantization
1	:	thresholding
2	:	scalar quantization
3	:	JPEG quantization (i.e., using tables)
4	:	Federal Bureau of Investigation quantization standard for fingerprints
5-7	:	reserved for other quantization modes (e.g., vector quantization)

Upon completion of quantization process 50, and decision 51 indicating that the desired compression ratio has been reached, quantization processor 62 presents its results to lossless compressor 64 for lossless compression process 62.

Lossless compressor 64 may be implemented by way of a conventional digital signal processor such as the TMS320C40 or TMS320C30 available from Texas Instruments Incorporated, the i860 processor available from Intel Corporation, or general purpose microprocessors such as the 80386 and 80486 available from Intel Corporation or the 68030 and 68040 available from Motorola, programmed in such a manner as to perform lossless compression process 52 upon the data from quantization processor 62. Lossless compression process 62 may be performed by lossless compressor 64 according to the desired conventional technique, such as Huffman encoding, adaptive Huffman encoding, arithmetic encoding, LSQ coding, and the like. Alternatively, lossless compressor 64 may be implemented as a custom logic circuit for providing this function.

The output of lossless compressor 64 is preferably compressed data in a bitstream format, and is presented to data flow interface 66. Data flow interface 66 provides an interface between spline-wavelet compressor

20 and network 26 or bus 25, and as such must arrange the bitstream output from lossless compressor 64 and arrange the same into a suitable format for transmission. Data flow interface 66 also provides a feedback signal to data controller 56 upon transmission of a frame of compressed data, based upon which data controller 56 may commence the processing of the next image to be compressed.

ii. Spline-wavelet decomposition

As noted above, according to this embodiment of the invention, spline-wavelet compressor 20 is operable to perform decomposition process 48 according to any one of three types of variations on a general spline-wavelet decomposition algorithms, namely dual-base wavelets, interpolatory wavelets, and wavelet packets. As each of the variations have different advantages relative to the others, the selection of a variation will depend upon the particular compression parameters for the input document to be compressed, and also will depend upon the decompression application (i.e., communication, archive retrieval, etc.). In summary, it is contemplated that dual-base wavelet decomposition will provide reasonable quality at reasonable computational speed, and as such will be useful as a default decomposition mode. It is contemplated that interpolatory wavelet decomposition will provide higher computational speed at a cost of reduced image reproduction quality, while wavelet packet decomposition will provide high quality and high compression ratio at a cost of slower execution speed.

Two of these three approaches, specifically the dual-base wavelets and interpolatory wavelets, utilize half-tree decomposition and reconstruction, while the wavelet packet approach utilizes full-tree decomposition.

Since a great deal of commonality exists between the two half-tree approaches, a generalized description of such half-tree decomposition using spline-wavelets techniques, encompassing both the dual-base wavelet and interpolatory wavelet approaches, will first be discussed.

(a) Half-tree spline-wavelet decomposition

In general, given an input signal function $f(x)$, a scaling (spline) function $\phi(x)$, and a wavelet function $\psi(x)$, one may project the input signal function $f(x)$ as a sequence of scaling function approximation coefficients $\{c_n^0\}$ associated with the spline function $\phi(x)$. These scaling function coefficients $\{c_n^0\}$ may be convolved with a sequence of scaling function decomposition coefficients $\{a_n\}$ to provide a low pass filter result, decomposing the projected scaling function coefficients $\{c_n^0\}$ into a blur, or low-frequency, component $\{c_n^{-1}\}$. Conversely, convolution of the approximation coefficient sequence $\{c_n^0\}$ with a sequence of wavelet function decomposition coefficients $\{b_n\}$ provides a band pass filter result, decomposing the projected scaling function approximation coefficients $\{c_n^0\}$ into a wavelet (or high frequency) component $\{d_n^{-1}\}$. This filtering and decomposition can then be repeated to the degree desired, such that the input signal function $f(x)$ is governed by approximation coefficient sequences $\{c_n^{-j}\}$, $\{d_n^{-j}\}$, where j is the level of decomposition.

Figure 5a illustrates a tree for the projection and decomposition of input signal $f(x)$ to two levels (i.e., $j = 2$). As illustrated in Figure 5a, the input signal function $f(x)$ is first projected into spline coefficients $\{c_n^0\}$. Convolution of spline approximation coefficient

sequence $\{c_n^0\}$ with decomposition sequences $\{a_n\}$, $\{b_n\}$ produces low-pass component approximation coefficient sequence $\{c_n^{-1}\}$ and band-pass component approximation coefficient sequence $\{d_n^{-1}\}$. Decomposition of low-pass spline approximation coefficient sequence $\{c_n^{-1}\}$ with decomposition sequences $\{a_n\}$, $\{b_n\}$ produces second-level low-pass and band-pass approximation coefficient sequences $\{c_n^{-2}\}$, $\{d_n^{-2}\}$, respectively.

For the two level case of Figure 5a, the input signal function f(x) can be represented by the approximation coefficient sequences $\{c_n^{-2}\}$, $\{d_n^{-1}\}$, and $\{d_n^{-2}\}$, assuming that one may reconstruct sequence $\{c_n^{-1}\}$ from sequences $\{c_n^{-2}\}$ and $\{d_n^{-2}\}$. Level-by-level reconstruction is illustrated in Figure 5b for this two-level case, beginning with the reconstruction of low pass component sequence $\{c_n^{-1}\}$ from sequences $\{c_n^{-2}\}$, $\{d_n^{-2}\}$ by convolution of the approximation coefficient sequences with reconstruction sequences $\{p_k\}$, $\{q_k\}$, respectively, and summing of the results. The next level involves the reconstruction of $\{c_n^0\}$, by convolution of approximation sequence components $\{c_n^{-1}\}$, $\{d_n^{-1}\}$ again with reconstruction sequences $\{p_k\}$, $\{q_k\}$, respectively, followed by summing of the results to arrive at approximation coefficient sequence $\{c_n^0\}$.

The efficiency and performance of any procedure for performing this decomposition and reconstruction, both for the methods according to the present invention and, in general, for all wavelet-based algorithms, depends upon the length and data type (i.e., integer, floating point, rational, irrational) of the decomposition sequences $\{a_n\}$, $\{b_n\}$ and the reconstruction sequences $\{p_k\}$, $\{q_k\}$. The values contained within these sequences depend primarily on the wavelet selected for use in the decomposition. In many cases, particularly those in the

prior art, the sequences $\{a_n\}$, $\{b_n\}$, $\{p_k\}$, $\{q_k\}$ are of infinite length and thus require truncation, and result in inefficient and inaccurate computation when attempted in conventional data processing equipment.

According to the preferred embodiment of the invention, the wavelet function selected for use in the decomposition is the compactly supported linear boundary spline wavelet or "boundary spline-wavelet" described in Chui and Quak, "Wavelets on a Bounded Interval", Numerical Methods of Approximation Theory, Volume 9 (Dec. 1992), pp. 53-75, incorporated herein by this reference. An advantageous feature of this wavelet is that its scaling function is a linear B-spline $\phi_2(x)$ having the following explicit formula:

$$\phi_2(x) = \begin{cases} x & x \text{ in } [0,1) \\ 2-x & x \text{ in } [1,2) \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Figure 6 graphically illustrates scaling function $\phi_2(x)$. This function has an extremely simple representation, and requires small support, such that any implementations including the evaluation of the wavelet, decomposition, and reconstruction, may be done effectively. Furthermore, it has been found that the linear B-spline satisfies the two-scale relation:

$$\phi_2(x) = \sum_{k=0}^{2} p_n \phi_2(2x-k)$$

where (2)

$$p_k = 2^{-m+1}\binom{m}{k} \tag{3}$$

For the case where m = 2, and k = {0, 1, 2}, $p_k = \frac{1}{2}\{1, 2, 1\}$.

For one-dimensional space, the linear spline wavelet function $\psi_2(x)$ may be explicitly expressed as:

$$\psi_2(x) = \sum_{k=0}^{4} q_k \phi_2(2x-k) \tag{4}$$

where:

$$q_k = (-1)^k 2^{-1} \sum_{l=0}^{2} \binom{2}{l} \phi_4(k+1-l),\ k=0,1,\ldots,4 \tag{5}$$

The expression for the linear spline wavelet $\psi_2(x)$ is in terms of the scaling function $\phi_2(x)$, which allows for relatively easy evaluation. These two scale relations of the linear B-spline and wavelets also provide a basis for general multiresolution analysis, which is utilized according to the preferred embodiment of the invention.

As is known in the art, any cardinal B-spline generates a multi-resolution analysis (MRA). Specifically, the collection of integer translates of a cardinal B-spline scaling function ϕ constitutes an approximation space V_0, using, for each integer j, the notation:

$$V_j = \{ \frac{\sum_k \alpha_k \phi(2^j x - k)}{\sum_k |\alpha_k|^2 < \infty} \} \tag{6}$$

and the notation $\phi(x) = N_m(x)$, for the m^{th} B-spline (usually with $m = 2$). For the scaling function ϕ to generate a multi-resolution analysis, the approximation spaces are related as follows:

$$\ldots\ V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \ldots$$

For each j, since $V_j \subset V_{j+1}$, there is a unique subspace W_j in V_{j+1} which is the orthogonal complement of V_j. This subspace W_j is called the wavelet subspace of V_{j+1}, represented by the notation:

$$V_{j+1} = V_j \oplus W_j \tag{7}$$

The approximation space V_0 is generated by the integer translates of scaling function ϕ; more generally, the j^{th} approximation space V_j is generated by $k/2^j$. Similarly, the wavelet subspace W_j is generated by $k/2^j$ translates of a spline-wavelet $\psi_m(x)$, namely:

$$W_j = \frac{\sum_k B_k \psi_m(2^j x - k)}{\sum_k |B_k|^2 < \infty} \tag{8}$$

Based upon the foregoing, according to the preferred embodiment of the invention, decomposition process 48 will now be generally described relative to Figure 6; note that certain options in the particular spline and

wavelet functions will be described in further detail hereinbelow. Process 48 begins with the projection of the input signal function f(x) into spline space, in process 68. In this example, it is preferred from the standpoint of computational simplicity to use the linear B-spline-wavelets, although higher order spline-wavelets may alternatively be used; it should be noted, however, that odd ordered spline-wavelets (i.e., linear, cubic, etc.) are preferably used according to the present invention.

According to the foregoing notation, the projection, or mapping, of the one-dimensional input function f(x) into spline space corresponds to the operation $f(x) \in V_m$. In the one-dimensional linear case, the input signal function f(x) maps to the function $f_0(x)$ in spline space V_0, which may be represented as follows:

$$f_0(x) = \sum_k c_k^0 \phi(x-k) \tag{9}$$

where k corresponds to the number of input signal samples over a selected interval. In the linear spline space V_0, the signal function $f_0(x)$ may be represented by the spline coefficients $\{c_k^0\}$. In the case of the linear B-spline, it is apparent that

$$f_0(k) = c_k^0 \tag{10}$$

giving rise to the properties that (i) the approximation coefficient sequence $\{c_k^0\}$ is exactly the original sequence of input signal samples, and (ii) the function $f_0(x)$ is exactly represented as the joining line segments between the input signal sample points.

Of course, the documents upon which the present invention is intended to operate are two-dimensional in nature; for example, the input signal function may be represented as $f_0(x,y)$. Process 68 of Figure 7 thus operates to project the input signal function into bilinear spline space. In this regard, consider a rectangle [a, b] by [c, d] to be partitioned into 2^{I+J} subrectangles of equal size, each subrectangle being represented as $[x_i, x_{i+1}]$ by $[y_j, y_{j+1}]$, where:

$$x_i = \frac{b-a}{2^I} i + a \quad , \quad i=0,1,\ldots,2^I$$

(11)

$$y_j = \frac{d-c}{2^J} j + c \quad , \quad j=0,1,\ldots,2^J$$

(12)

As a result, the rectangle is represented by $(2^I+1)(2^J+1)$ equally spaced grid points (x_i, y_j), for i from 0 to 2^I, and for j from 0 to 2^J over the rectangle [a, b] by [c, d].

In process 68, a sampled input signal sequence $f(x_i, y_j)$ over the range of i from 0 to 2^I, and j from 0 to 2^J, is received in PGM format from format converter 12 by DSP 60 in spline-wavelet compressor 20. In the general case, process 68 projects the sampled inputs signal sequence $f(x_i, y_j)$ by application of a bilinear spline function into a sequence $(S_{I,J}f)(x,y)$ as follows: